DW03_updn_ctr

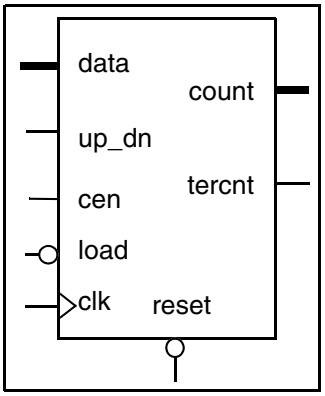
Up/Down Counter

Version, STAR, and myDesignWare Subscriptions: IP Directory

Features and Benefits

- Up/down count control
- Asynchronous reset
- Loadable count register
- Counter enable
- Terminal count flag

Revision History



Description

DW03_updn_ctr is a general-purpose binary up-down counter. The counter data path is *width* bits wide and has 2^{width} states from "000...000" to "111...111", depending on the specified width. The counter is clocked on the positive edge of the `clk` input.

Table 1-1 Pin Description

Pin Name	Width	Direction	Function
data	*width* bits	Input	Input data bus
up_dn	1 bit	Input	Count up (*up_dn* = 1) or count down (*up_dn* = 0)
load	1 bit	Input	Counter load enable, active low
cen	1 bit	Input	Counter enable, active high
clk	1 bit	Input	Clock
reset	1 bit	Input	Asynchronous counter reset, active low
count	*width* bits	Output	Output count bus
tercnt	1 bit	Output	Terminal count flag

Table 1-2 Parameter Description

Parameter	Value	Function
width	≥ 1	Width of the `count` output bus

Table 1-3 Synthesis Implementations

Implementation Name	Function	License Feature Required
str	Synthesis model	DesignWare

Table 1-4 Simulation Models

Model	Function
DW03.DW03_UPDN_CTR_CFG_SIM	Design unit name for VHDL simulation
dw/dw03/src/DW03_updn_ctr_sim.vhd	VHDL simulation model source code
dw/sim_ver/DW03_updn_ctr.v	Verilog simulation model source code

Table 1-5 Counter Operation Truth Table

reset	load	cen	up_dn	Operation
0	X	X	X	Reset
1	0	X	X	Load
1	1	0	X	Standby
1	1	1	0	Count down
1	1	1	1	Count up

The `reset`, active low, provides for an asynchronous reset of the counter to “000...0”. If the reset pin is connected to ‘1’, then the reset logic is not synthesized, resulting in a smaller and faster counter.

The `up_dn` input controls whether the counter counts up (`up_dn` is high) or down (`up_dn` is low), starting on the next `clk` cycle.

The counter is loaded with data by asserting `load` (low) and applying data on the `data` input. The data load operation is synchronous with respect to the positive edge of `clk`.

The count enable pin, `cen`, is active high. When `cen` is high, the counter is active. When `cen` is low, the counter is disabled and `count` remains at the same value.

The `tercnt` is an output port. When counting up, `tercnt` is high at `count` = “111....111”. When counting down, `tercnt` is high at `count` = “000....000”.

Suppressing Warning Messages During Verilog Simulation

The Verilog simulation model includes macros that allow you to suppress warning messages during simulation.

To suppress all warning messages for all DWBB components, define the DW_SUPPRESS_WARN macro in either of the following ways:

- Specify the Verilog preprocessing macro in Verilog code:

  ```
  `define DW_SUPPRESS_WARN
  ```

- Or, include a command line option to the simulator, such as:

  `+define+DW_SUPPRESS_WARN` (which is used for the Synopsys VCS simulator)

The warning messages for this model include the following:

- If values other than 1 or 0 are present on a clock port, the following message is displayed:

  ```
  WARNING: <instance_path>.<clock_name>_monitor:
      at time = <timestamp>, Detected unknown value, x, on <clock_name> input.
  ```

  To suppress only this warning message for all DWBB components, use the following macro:

  - Define the DW_DISABLE_CLK_MONITOR macro. You can define this macro in the following ways:
    - Specify the Verilog preprocessing macro in Verilog code:

      ```
      `define DW_DISABLE_CLK_MONITOR
      ```

    - Or, include a command line option to the simulator, such as:

      `+define+DW_DISABLE_CLK_MONITOR` (which is used for the Synopsys VCS simulator)

  This message is also suppressed using the DW_SUPPRESS_WARN macro explained earlier.

Timing Diagrams

The following timing diagrams show various conditions for DW03_updn_ctr.

Figure 1-1 Functional Operation - 1

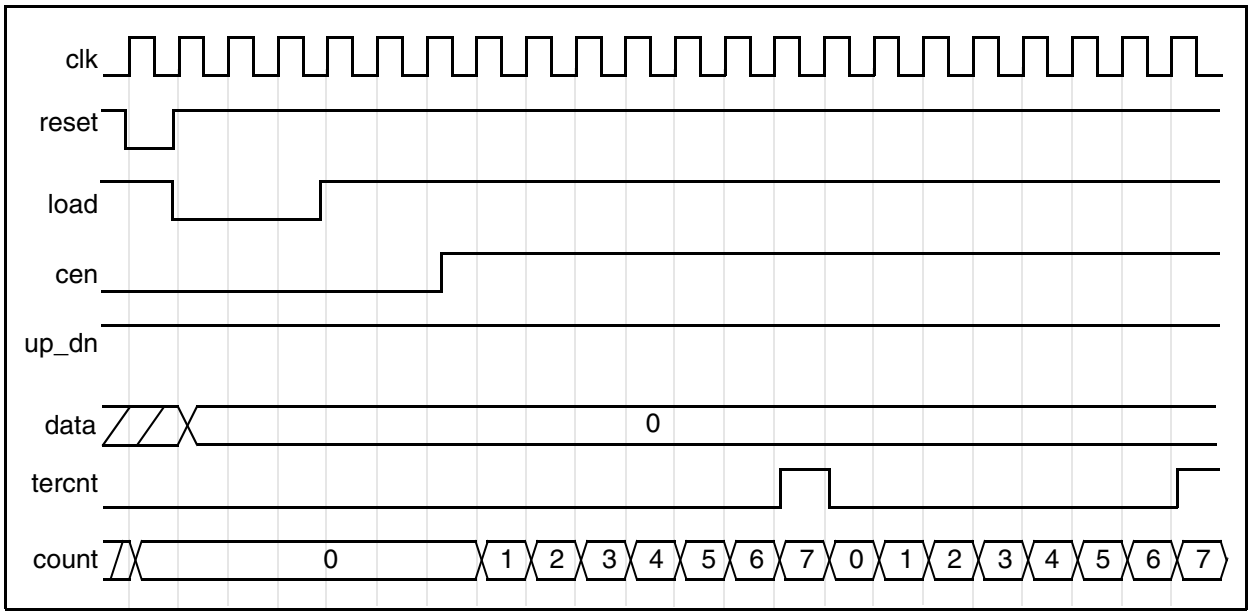
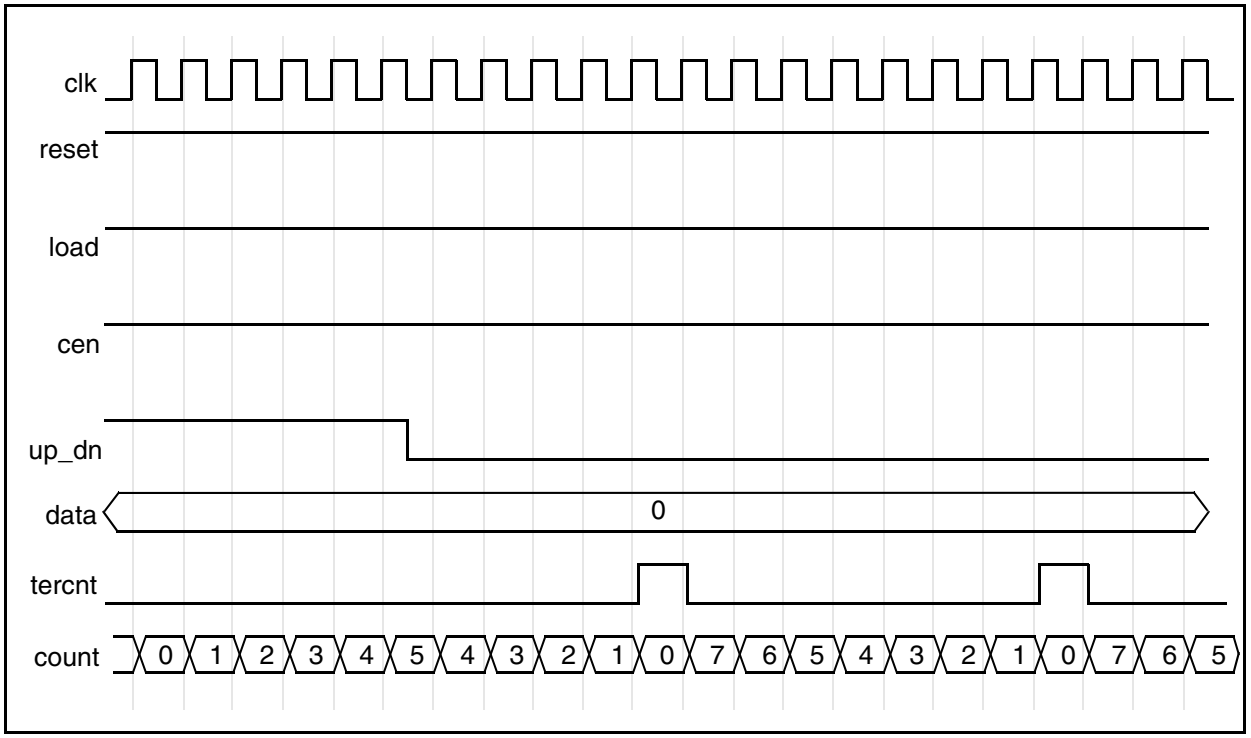


Figure 1-2 Functional Operation - 2



Related Topics

- Logic – Sequential Overview
- DesignWare Building Block IP User Guide

HDL Usage Through Component Instantiation - VHDL

```
library IEEE,DWARE;
use IEEE.std_logic_1164.all;
use DWARE.DWpackages.all;
use DWARE.DW_foundation_comp.all;

entity DW03_updn_ctr_inst is
  generic ( inst_width : POSITIVE := 8 );
  port ( inst_data   : in std_logic_vector(inst_width-1 downto 0);
         inst_up_dn  : in std_logic;
         inst_load   : in std_logic;
         inst_cen    : in std_logic;
         inst_clk    : in std_logic;
         inst_reset  : in std_logic;
         count_inst  : out std_logic_vector(inst_width-1 downto 0);
         tercnt_inst : out std_logic );
end DW03_updn_ctr_inst;

architecture inst of DW03_updn_ctr_inst is
begin

  -- Instance of DW03_updn_ctr
  U1 : DW03_updn_ctr
    generic map ( width => inst_width )
    port map ( data => inst_data, up_dn => inst_up_dn,
               load => inst_load, cen => inst_cen, clk => inst_clk,
               reset => inst_reset, count => count_inst,
               tercnt => tercnt_inst );
end inst;

-- pragma translate_off
configuration DW03_updn_ctr_inst_cfg_inst of DW03_updn_ctr_inst is
  for inst
  end for; -- inst
end DW03_updn_ctr_inst_cfg_inst;
-- pragma translate_on
```

HDL Usage Through Component Instantiation - Verilog

```
module DW03_updn_ctr_inst( inst_data, inst_up_dn, inst_load,
                           inst_cen, inst_clk, inst_reset,
                           count_inst, tercnt_inst );

  parameter width = 8;

  input [width-1 : 0] inst_data;
  input inst_up_dn;
  input inst_load;
  input inst_cen;
  input inst_clk;
  input inst_reset;
  output [width-1 : 0] count_inst;
  output tercnt_inst;

  // Instance of DW03_updn_ctr
  DW03_updn_ctr #(width)
    U1 ( .data(inst_data), .up_dn(inst_up_dn), .load(inst_load),
         .cen(inst_cen), .clk(inst_clk), .reset(inst_reset),
         .count(count_inst), .tercnt(tercnt_inst) );

endmodule
```

Revision History

For notes about this release, see the *DesignWare Building Block IP Release Notes*.

For lists of both known and fixed issues for this component, refer to the STAR report.

For a version of this datasheet with visible change bars, click here.

Date	Release	Updates
July 2020	DWBB_201912.5	■ Adjusted content and title of "Suppressing Warning Messages During Verilog Simulation" on page 3 and added the DW_SUPPRESS_WARN macro
October 2019	DWBB_201903.5	■ Added the "Disabling Clock Monitor Messages" section
March 2019	DWBB_201903.0	■ Removed minPower designation from this datasheet
January 2019	DWBB_201806.5	■ Updated example in "HDL Usage Through Component Instantiation - VHDL" on page 5 ■ Added this Revision History table and the document links on this page

Copyright Notice and Proprietary Information

Destination Control Statement

All technical data contained in this publication is subject to the export control laws of the United States of America. Disclosure to nationals of other countries contrary to United States law is prohibited. It is the reader's responsibility to determine the applicable regulations and to comply with them.

Disclaimer

SYNOPSYS, INC., AND ITS LICENSORS MAKE NO WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, WITH REGARD TO THIS MATERIAL, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE.

Trademarks

Synopsys and certain Synopsys product names are trademarks of Synopsys, as set forth at https://www.synopsys.com/company/legal/trademarks-brands.html.

All other product or company names may be trademarks of their respective owners.

Free and Open-Source Software Licensing Notices

If applicable, Free and Open-Source Software (FOSS) licensing notices are available in the product installation.

Third-Party Links

Any links to third-party websites included in this document are for your convenience only. Synopsys does not endorse and is not responsible for such websites and their practices, including privacy practices, availability, and content.

Synopsys, Inc.
www.synopsys.com